* **AI Assignment - Part 1: Theoretical Understanding (40%)**

### 1. Short Answer Questions

**Q1: Explain the primary differences between TensorFlow and PyTorch. When would you choose one over the other?**

**Answer:** TensorFlow and PyTorch are both popular deep learning frameworks.

* **TensorFlow** uses static computation graphs (though Eager Execution allows dynamic behavior) and is widely used in production environments, especially with TensorFlow Serving and TensorFlow Lite.
* **PyTorch** uses dynamic computation graphs, which makes it more intuitive and pythonic for debugging and experimentation.

**When to choose:**

* Use **TensorFlow** for production-level deployment, mobile/embedded AI, and compatibility with tools like TensorFlow Extended.
* Use **PyTorch** for fast prototyping, academic research, and projects needing dynamic graphs.

**Q2: Describe two use cases for Jupyter Notebooks in AI development.**

**Answer:**

1. **Interactive Data Exploration:** Jupyter allows developers to load datasets, visualize them with plots, and clean data step-by-step while documenting the process.
2. **Model Prototyping and Testing:** It helps test various ML/DL models in cells, visualize training progress, and adjust parameters on the fly.

**Q3: How does spaCy enhance NLP tasks compared to basic Python string operations?**

**Answer:** spaCy provides advanced NLP features such as tokenization, part-of-speech tagging, named entity recognition, and dependency parsing. Unlike basic Python string operations (which can only split, replace, or search text), spaCy understands the context and structure of language, making it ideal for real-world text analysis.

### 2. Comparative Analysis

**Scikit-learn vs. TensorFlow**

|  |  |  |
| --- | --- | --- |
| Feature | Scikit-learn | TensorFlow |
| Target Applications | Classical ML (e.g., SVM, Decision Trees) | Deep Learning (CNNs, RNNs, Transformers) |
| Ease of Use | Very beginner-friendly, concise API | Steeper learning curve, more setup needed |
| Community Support | Strong, especially in academia | Massive, especially in industry and production |

**Conclusion:** Use **Scikit-learn** for simpler problems and structured data. Choose **TensorFlow** when working with neural networks or complex deep learning tasks.

End of Part 1 Document.